

Documentação de resultados usando R Markdown

Vanderlei Júlio Debastiani (vanderleidebastiani@yahoo.com.br)

05 Janeiro 2019

Índice de conteúdo

1	Introdução	1
2	Estrutura do RMD	1
2.1	Cabeçalho YAML	1
2.2	Texto formatado com R Markdown	2
2.2.1	Texto simples	2
2.2.2	Formatação de texto básica	2
2.2.3	Cabeçalhos	2
2.2.4	Equações	2
2.2.5	Links	2
2.2.6	Listas	2
2.3	Códigos R (usando chunks)	3
2.4	Gráficos	4
2.5	Tabelas	5
3	Exportar o arquivo <i>.rmd</i> nos formatos de saída	6
4	Conclusão	6
5	Mais informações	6
6	Referências	7

1 Introdução

A documentação de resultados é uma etapa importante no desenvolvimento da ciência. R Markdown é uma ferramenta para produzir saídas formatadas de resultados do R, tornando as rotinas de análises e resultados mais simples e fácil de serem registrados e posteriormente compartilhados. O ponto de partida é um arquivo *.rmd*. Este arquivo é usado como um script do R, onde os comandos podem ser editados, comentados e executados normalmente. Usando um conjunto simplificado de sintaxe é possível adicionar textos, figuras, tabelas, links e outros elementos para complementar a documentação dos resultados. Por fim, os resultados podem ser facilmente formatados e exportados para vários formatos, incluindo HTML, PDF, RTF, MS Word e LaTeX. O objetivo deste tutorial é mostrar as opções básicas para documentação de resultados, usando como exemplo as saída no formato PDF.

2 Estrutura do RMD

O arquivo *.rmd* é composto por diferentes seções, este arquivo é usado na função *render* do pacote *rmarkdown* para converter no formato de saída especificado.

2.1 Cabeçalho YAML

A primeira seção é cabeçalho YAML (opcional) usada na função *render*. Argumentos de personalização das saída podem ser especificadas no cabeçalho ou diretamente na função *render*.

Essa seção começa com `---` e termina com `---`.

```

---
title: "Título"
author: "Nome"
date: "Data"
---
```

2.2 Texto formatado com R Markdown

O texto na saída pode ser formatado usando um conjunto simples de sintaxe. Algumas das opções básicas estão listados abaixo:

2.2.1 Texto simples

Os textos simples podem ser usados sem qualquer sintaxe. Para iniciar um novo parágrafo é possível usar dois espaços no final ou deixar uma linha em branco.

2.2.2 Formatação de texto básica

```

*itálico* para itálico;
**negrito** para negrito;
‘códigos’ para código textual;
superscrito^2~ para superscrito2;
subescrito~2~ para subescrito2;
barra invertida para ignorar caracter especial *, ‘ e \;
> bloco de citação para > bloco de citação;
```

2.2.3 Cabeçalhos

Os cabeçalhos são especificados usando o #.

```

# Cabeçalhos 1
## Cabeçalhos 2
### Cabeçalhos 3
#### Cabeçalhos 4
##### Cabeçalhos 5
```

2.2.4 Equações

Equações são especificadas usando \$ ou equações em bloco usando \$\$:

```

$A = \pi*r^{2}$ para  $A = \pi * r^2$ ;
$$A = \pi*r^{2}$$ para
```

$$A = \pi * r^2$$

2.2.5 Links

```

<http://www.google.com> para http://www.google.com
[Site de Busca](www.google.com) para Site de Busca
```

2.2.6 Listas

Para listas não ordenadas são definidas pelos caracteres *, + e -:

```

* item 1
+ sub-item 1
+ sub-item 2
- sub-sub-item 1
* item 2
```

- item 1
 - sub-item 1
 - sub-item 2
 - * sub-sub-item 1
- item 2

Para listas ordenadas:

1. item 1
2. item 2
 - i) sub-item 1
 - A. sub-sub-item 1

1. item 1
2. item 2
 - i) sub-item 1
 - A. sub-sub-item 1

2.3 Códigos R (usando chunks)

Para inserir códigos de R, como scripts e resultados, é possível usar a sintaxe chunk que começa com ‘ ‘ { r } e termina com ‘ ‘ ‘. Uma ou mais linhas podem ser incluídas no mesmo bloco.

Exemplo:

```
““{r}
data(C02)
head(C02)
summary(C02[,4:5])
““
```

```
data(C02)
head(C02)
```

```
##   Plant   Type Treatment conc uptake
## 1  Qn1 Quebec nonchilled   95   16.0
## 2  Qn1 Quebec nonchilled  175   30.4
## 3  Qn1 Quebec nonchilled  250   34.8
## 4  Qn1 Quebec nonchilled  350   37.2
## 5  Qn1 Quebec nonchilled  500   35.3
## 6  Qn1 Quebec nonchilled  675   39.2
```

```
summary(C02[,4:5])
```

```
##           conc           uptake
## Min.      : 95      Min.      : 7.70
## 1st Qu.: 175      1st Qu.:17.90
## Median : 350      Median :28.30
## Mean     : 435      Mean     :27.21
## 3rd Qu.: 675      3rd Qu.:37.12
## Max.     :1000     Max.      :45.50
```

Chunk incluem algumas opções para a formatação do bloco de códigos e para exibir os resultados. Os argumentos são especificados dentro do primeiro par de chaves {r, highlight = FALSE }. Algumas dessas opções:

- `eval` - Avaliar validade do código (default `eval = FALSE`);
- `echo` - Mostrar código no documento de saída (default `= TRUE`);
- `collapse` - Recolher todos saída em bloco único (default `= FALSE`);
- `results` (default `= 'markup'`):
 - `'markup'` - mostrar os resultados destacados no meio do código;
 - `'hide'` - não mostrar os resultados;
 - `'hold'` - colocar todos os resultados abaixo de todo o código;
 - `'asis'` - mostrar os resultados brutos no meio do código.
- `message` - Mostrar mensagens no código no documento de saída (default `= TRUE`);
- `warning` - Mostrar mensagens de warnings no documento de saída (default `= TRUE`);
- `highlight` - Descatar código fonte (default `= TRUE`);
- `fig.height`, `fig.width` - Dimensões das figuras em polegadas.

Exemplo:

```
“‘{r, highlight = FALSE}
# Show R version
getRversion()
“‘
```

```
# Show R version
getRversion()
```

```
## [1] '3.5.2'
```

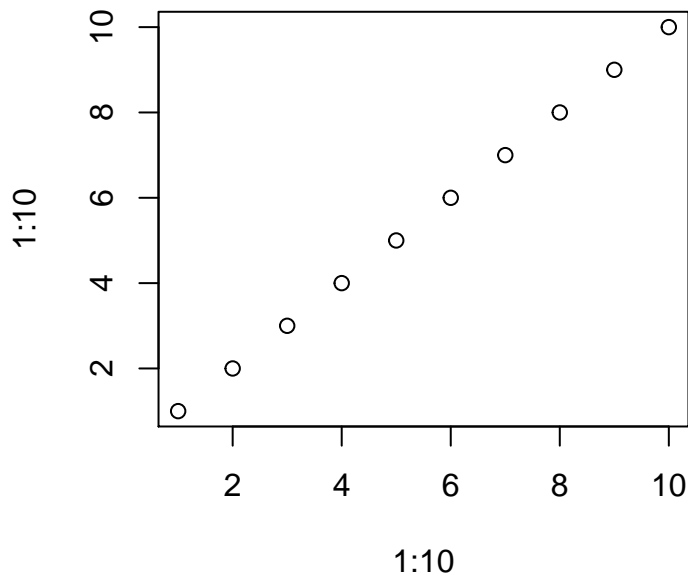
2.4 Gráficos

Os resultados gráficos podem ser incluídos no documento com os resultado da saída, exatamente como as funções de *plot* do R. Por padrão, os tamanhos das figuras são especificados no cabeçalho YAML ou na função *render*, mas também os tamanhos podem ser especificados para cada gráfico usando as opções do chunk.

Exemplo:

```
“‘{r, fig.width = 4, fig.height = 4}
plot(1:10,1:10)
“‘
```

```
plot(1:10, 1:10)
```



2.5 Tabelas

As tabelas podem ser incluídas usando a opção de impressão (*print*) nos códigos R (código chunks) ou usando as funções para formatar dados em tabelas disponíveis no R. Alguns exemplos são as funções *kable* do pacote *knitr*, *xtable* do pacote *xtable* e a função *stargazer* do pacote *stargazer*.

Exemplo:

```
“{r}
require(knitr)
C02[1:10,]
knitr::kable(C02[1:10,], caption = "Tabela com kable")
“
```

```
require(knitr)
```

```
## Loading required package: knitr
```

```
C02[1:10,]
```

```
##      Plant  Type Treatment conc uptake
## 1   Qn1 Quebec nonchilled   95   16.0
## 2   Qn1 Quebec nonchilled  175   30.4
## 3   Qn1 Quebec nonchilled  250   34.8
## 4   Qn1 Quebec nonchilled  350   37.2
## 5   Qn1 Quebec nonchilled  500   35.3
## 6   Qn1 Quebec nonchilled  675   39.2
## 7   Qn1 Quebec nonchilled 1000   39.7
## 8   Qn2 Quebec nonchilled   95   13.6
## 9   Qn2 Quebec nonchilled  175   27.3
## 10  Qn2 Quebec nonchilled  250   37.1
```

```
knitr::kable(C02[1:10,], caption = "Tabela com kable")
```

Tabela 1: Tabela com kable

Plant	Type	Treatment	conc	uptake
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2
Qn1	Quebec	nonchilled	1000	39.7
Qn2	Quebec	nonchilled	95	13.6
Qn2	Quebec	nonchilled	175	27.3
Qn2	Quebec	nonchilled	250	37.1

3 Exportar o arquivo *.rmd* nos formatos de saída

A função *render* é usada para exportar o arquivo *.rmd* no documento final formatado. A função carrega o arquivo e converte para o formato de saída especificado no cabeçalho YAML ou diretamente na função. As opções para a saída são descritas na funções *pdf_document*, *html_document*, *rtf_document* e *word_document*. Vários argumentos são suportados para cada formato de saída. Algumas das opções para *pdf_document*:

- `number_sections` - Adicionar seção de numeração para cabeçalhos (default `number_sections = FALSE`);
- `fig_caption` - Processar figuras com legendas (default `fig_caption = TRUE`);
- `fig_height`, `fig_width` - Altura e largura padrão para as figuras (em polegadas) para o documento (default `fig_width = 6.5` e `fig_height = 4.5`);
- `highlight` - Destaque de sintaxe dos códigos: “tango”, “pygments”, “kate”, “zenburn”, “textmate” (default `highlight = default`);
- `keep_tex` - Salvar uma cópia do arquivo *.tex* (default `keep_tex = TRUE`).

Exemplo:

```
require(rmarkdown)
require(knitr)
render("input.Rmd", pdf_document(keep_tex = TRUE))
```

4 Conclusão

O objetivo deste texto foi introduzir conceitos básicos para documentação de resultados usando R Markdown. Espero que este texto tenha sido útil e, por favor, avise-me se tiver dúvidas ou sugestões sobre este texto.

5 Mais informações

Outros textos e tutoriais sobre R podem ser encontrados em <https://vanderleidebastiani.github.io/tutoriais>.

6 Referências

- R Core Team. 2018. **R Language Definition**. <https://cran.r-project.org/doc/manuals/R-lang.html>
- R Markdown. 2018. **R Markdown**. <http://rmarkdown.rstudio.com>
- Knitr. 2018. **knitr - Elegant, flexible, and fast dynamic report generation with R**. <https://yihui.name/knitr/>